DOCUMENT RESUME

ED 302 221 IR 013 583

AUTHOR Jones, Preston K.
TITLE The Effect of Computer Programming Instruction on the Development of Generalized Problem Solving Skills in High School Students.
PUB DATE 14 Jun 88
NOTE 46p.; Ed.D. Practicum, Nova University.
PUB TYPE Dissertations/Theses - Practicum Papers (043)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS *Cognitive Processes; Comparative Analysis; Computer Science Education; *Critical Thinking; High Schools; *High School Students; Hypothesis Testing; Problem Solving; *Programing; *Skill Development
IDENTIFIERS *BASIC Programing Language; *PASCAL Programing Language

ABSTRACT

This study investigated the extent to which participation by high school students in a computer programming course leads to improved critical thinking skills, and whether a significant difference exists in critical thinking skill development between participants in Pascal and BASIC classes. Three groups of high school students were tested using the Watson-Glaser Critical Thinking Appraisal: participants in a first-year BASIC class (N=52), participants in a first-year Pascal class (N=21), and above average (N=40). students in other classes who had had no exposure to programming. Group measures were compared using the t-test. Students enrolled in both programming classes scored significantly higher than their non-programming counterparts; however, exposure to a specific programming language--BASIC or Pascal--did not produce significant differences in test scores. The text is supplemented by five tables, and a copy of the survey instrument used to gather opinions on this topic from members of the FICS (Florida Instructional Computing Supervisors) is appended. (37 references) (EW)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED302221

PRACTICUM REPORT

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

Center for the
Advancement of Education
Ed.D. Program

THE EFFECT OF COMPUTER PROGRAMMING INSTRUCTION ON THE DEVELOPMENT OF GENERALIZED PROBLEM SOLVING SKILLS IN HIGH SCHOOL STUDENTS

IR013593

301 College Avenue
ort Lauderdale, Florida 33314
05/475-7300

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Preston K. Jones

TO THE EDU[illegible]IONAL RESOURCES INFORMATION CENTER (ERIC)."

NOVA UNIVERSITY

THE EFFECT OF COMPUTER PROGRAMMING INSTRUCTION ON THE DEVELOPMENT OF GENERALIZED PROBLEM SOLVING SKILLS IN HIGH SCHOOL STUDENTS

by

PRESTON K. JONES
(Jonesp)

A PRACTICUM (MAP) REPORT SUBMITTED TO NOVA UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF DOCTOR OF EDUCATION IN
COMPUTER EDUCATION

JUNE 14, 1988

TABLE OF CONTENTS

SECTION:	PAGE#:
TITLE PAGE	I
TABLE OF CONTENTS	II
ABSTRACT	III
LIST OF TABLES	v
CHAPTER	
I. Introduction.	1
II. Problem Statement	3
III. Research Hypotheses	4
IV. Background	5
V. Definition of Terms	11
VI. Limitations of the Study.	13
VII. Basic Assumptions	15
VIII. Data Collection Procedures.	16
IX. Data Treatment Procedures	18
X. Presentation of Results	20
XI. Conclusions and Significance.	27
XII. Further Studies	32
REFERENCE LIST	34
APPENDIXES	
A. Computer Language Survey	38

ABSTRACT

Most computer educators advocating programming in high school point to development of critical thinking/problem-solving skills as their reason for support. Generally, use of a structured language, such as Pascal, is recommended because of modularization and variable declaration. However, examination of current literature reveals that little empirical evidence is available to support the position. Existing research generally has centered on elementary and middle school students.

The purpose of this study was to gather and examine data, to determine empirically the extent to which participation by high school students in a programming course leads to improved critical thinking skills, and whether a significant difference exists in critical thinking skill development between participants in Pascal and BASIC classes.

Three groups of high school students were tested using the Watson-Glaser Critical Thinking Appraisal: participants in a first-year BASIC class, participants in a first-year Pascal class, and above average students in other classes having had no exposure to programming. Group means were compared using the t-test.

Students enrolled in programming classes (despite language) scored significantly higher on the Watson-Glaser than their non-programming counterparts. However, exposure to a specific language (Pascal or BASIC) did not produce any significant difference in test scores. Analysis of the five subtests on the Watson-Glaser produced results sufficiently interesting to suggest a need for further study.

LIST OF TABLES

Table		Page:
1.	A Summary of Scores for the BASIC Programming Students on the Watson-Glaser Critical Thinking Appraisal	21 & 22
2.	A Summary of Scores for the Pascal Programming Students on the Watson-Glaser Critical Thinking Appraisal	23
3.	A Summary of Scores for the Non-programming Students on the Watson-Glaser Critical Thinking Appraisal	24 & 25
4.	A Summary of Descriptive Statistics for the Programming Students on the Watson-Glaser Critical Thinking Appraisal	25
5.	A Summary of t-ratios Comparing the Means of the Programming, BASIC, Pascal and Non-programming Groups.	26

I. INTRODUCTION

Computer educators generally agree: Programming should be part of a school or district curriculum, but concede that less than one or two percent of the future work force will consist of computer programmers (Luehrmann, 1985). Advocates point to such concepts as development of logical processes, critical thinking skills, and effective problem-solving techniques as reasons for support (Burnett & Friesen, 1986). The topic has been discussed frequently by members of the Florida Instructional Computing Supervisors (FICS). As President of this group, the author has been privy to many such conversations. A November, 1987 sampling (See Appendix A) of the FICS membership revealed that 71% favor programming because said activity develops problem solving/critical thinking skills. Most prefer Pascal over BASIC, because of its procedural characteristics. As Cynthia Solomon (1982:173) explains, "the powerful problem-solving strategy of breaking problems into smaller and smaller parts can only be a paper-and-pencil strategy in a BASIC programming environment." Such is not the case in Pascal. Brian Harvey (1983:168) is more direct, "all respectable languages are procedural, by definition."

Some researchers such as Clements and Gullo (1984), have attempted to determine the effects of Logo programming on development of cognitive and problem solving skills in young children. Unfortunately, few such projects have been undertaken to determine like effects in high school students. While most FICS members, and others, maintain the position that exposure to programming courses (particularly Pascal) leads to improved cognitive and problem-solving ability, empirical evidence to support such a position is lacking (Dalbey & Linn, 1985).

II. PROBLEM STATEMENT

The problem addressed is a lack of empirical evidence to support (1) teaching of programming at the high school level for its problem-solving benefits, and (2) the extent to which programming in BASIC or Pascal environments has a significantly greater influence on the development of problem-solving skills.

III. RESEARCH HYPOTHESES

Four hypotheses were investigated.

1. Students participating in computer programming classes (despite language) will not score significantly higher on the Watson-Glaser Critical Thinking Appraisal, when compared with students who have not experienced a programming environment.
2. Students participating in BASIC programming classes will not score significantly higher on the Watson-Glaser Critical Thinking Appraisal, when compared with students who have not experienced a programming environment.
3. Students participating in Pascal programming classes will not score significantly higher on the Watson-Glaser Critical Thinking Appraisal, when compared with students who have not experienced a programming environment.
4. On the Watson-Glaser Critical Thinking Appraisal, there will be no significant difference between the scores of students participating in a BASIC programming class and those participating in a Pascal programming class.

IV. BACKGROUND

There has been tremendous impetus in recent years to improve critical thinking skills. Current educational literature abounds with the importance of developing said skills (Sternberg, 1985). Many national commissions, state education agencies, and professional educators have extolled the virtue of teaching thinking skills at all levels of schooling (Beyer and Charlton, 1986). A major goal of education must be to teach students to think critically and independently. Hudgins (1977:173) defines critical thinking as "the student's ability to attend carefully and to comprehend the meaning of material." The fact that much energy and concern are being focused on providing better techniques for teaching thinking skills cannot be denied (Beyer, 1984). The question is how best to provide instruction that will lead to development of effective problem-solving skills.

Margaret Perry (1986), and others, believe a partial solution is to teach computer programming. As James Stronge (1987:19) states,

> Programming is an activity that requires students to engage extensively in higher-level cognitive skills. As students begin to design and code a program, they rely heavily on the cognitive skills of analysis, synthesis and evaluation.

Arthur Luehrmann (1985) and Seymour Papert (1980) are outspoken advocates of programming. They believe such experiences provide students with a significant way to think about themselves and the world around them. The programming student is viewed as an "epistemologist": a problem solver, able to learn about learning through participation.

Proponents of programming as a means of developing problem-solving skills emphasize the importance of using a structured language. Said usage helps students understand that problem solving is a process of defining a goal, developing a plan, and organizing a step-by-step solution (Clements, 1983; Shelly, Cashman & Forsythe, 1987; Burnett & Friesen, 1986; Markuson, Tobias, & Lough, 1983; Schank, 1984). When speaking of the value of Pascal as a structured language, Presley and Corica (1986:1) write,

> It is our belief that learning to program offers the student an invaluable opportunity to develop problem-solving skills. The process of defining a problem, breaking it down into a series of smaller problems and finally writing a computer program to solve it, is an exercise in learning to think logically.

Pascal does not allow the programmer to extemporize and produce what is frequently called "spaghetti code." Variables must be declared at the outset and programs subdivided into modules called functions and procedures. According to David Eckerman

(1983), the greatest limitation to BASIC is production of programs that are error-prone because all variables are declared globally. Also, the meandering nature of the programs renders them impossible to decipher at a later time.

Some research has shown that programming leads to increased development of critical-thinking/problem-solving skills. In a study of six-year-old children, Clements and Gullo (1984) demonstrated that those exposed to computer programming experiences scored significantly higher on tests of impulsivity/reflectivity and divergent thinking (important abilities for effective critical thinking) than did their counterparts who were treated with exposure to Computer Assisted Instruction only. Marcia Linn (1985) in a study of the cognitive consequences of programming instruction in classrooms concluded,

> . . . introductory courses in programming, compared to other introductory courses in, say, French or physics have great potential for emphasizing problem solving. Specifically, the computer learning environment offers more precise and interactive feedback on problem solving than is found in most other environments. Students can use on-line feedback to diagnose when their problem solutions fail and to determine how well various solutions to problems work without the help of the teacher.

Clements (1987), in a long-term study of elementary students, found evidence that Logo programming leads to

increased ability to solve analogies and sequencing problems.

While the foregoing is convincing, there is ample support for the opposite view: programming experiences do not result in increased development of problem-solving/critical-thinking skills. The abilities developed through programming are designed to generalize to other disciplines. However, recent research by cognitive scientists has demonstrated that learning is more domain-specific than first thought. Clearly, the teaching of programming does not necessarily generalize problem-solving skills to other disciplines. Great care must be exercised to structure a course in a carefully chosen language for generalization to occur (Linn, 1985; Dalbey & Linn, 1985).

Pea and Kurland (in Pintrich, Berger & Stemmer, 1987) found little evidence to support the proposition that programming leads to higher-order thinking skills. In a study of Logo programming, no significant difference was found between Logo and control groups on the development of planning skills. Seymour Papert (1987) argues that this and similar studies comparing control and experimental Logo groups are much too narrow in focus. Researchers need to look at a broad

spectrum of thinking skills and not concentrate on one or two.

Clearly, little agreement exists regarding the value of teaching computer programming to develop critical thinking skills. Also, it is clear that the same disagreement pervades the choice of language. BASIC is the most predominant (Hughes, 1983) and preferred language in use today (Staff, 1987). However, programming purists, such as the members of FICS, prefer structured languages like Pascal. Frequently the question is asked: How can all this divergence be explained?

In the view of this author, a partial answer is that most researchers and theorists have failed to consider the work of Jean Piaget in developing studies, interpreting results, and proffering postulates. Piaget identified four stages of cognitive development.

1. From birth to two years the child is in the sensorimotor stage, characterized by simple reflexes and beginning habits.
2. The pre-operational stage engenders ages two to seven. The mental capacity of the child is characterized by static images.
3. The years seven through twelve find the child at the concrete operational stage. Here s/he masters the concept of conservation and can

perform logical operations with real objects, subject to manipulation.

4. Last, at age 12 and beyond, the child can achieve formal operations. Reasoning beyond the here and now, and conceptualization of alternative solutions to a problem become evident. (Hilgard, Atkinson & Atkinson, 1975; Hudgins, 1977; Ginsberg & Opper, 1969; Flavell, 1963).

Effective programming requires that the student: (a) understand the problem, (b) design and plan a solution (program), (c) code, and (d) debug the program (Pea and Kurland in Pintrich et al., 1987). Surely, Piaget would classify programming as a formal operations activity. However, most empirical research has concentrated on elementary and middle school students, largely not at the formal operations level of development. It seems clear that most research has targeted the wrong group.

A notable exception is the work of Paul Pintrich, Carl Berger, and Paul Stemmer. They examined the behavior of programming students in a high school Pascal course. The study relied on frequency of observed phenomena. They concluded that "students did not engage frequently in behavior that might be indicative of on-going higher level cognitive processes" (Pintrich et al., 1987:460). The fallacy: Development of cognitive processes were not measured but inferred from participant behavior.

V. DEFINITION OF TERMS

BASIC--Beginners All-purpose Symbolic Instruction Code. A non-structured high level computer programming language. It is the most commonly used language today.

Code--A series of instructions, in a specific language, issued to a computer to solve a problem.

(Computer) Program--A series of step by step instructions, written in a particular language, that a computer can use to provide a solution to a problem.

Curriculum--All learning opportunities provided by a school or school system (Saylor & Alexander, 1966).

Elementary School--Grades K-5.

Epistemology--The theory of the nature of knowledge.

High School--Grades 9-12.

Logo--A programming language developed for children by Seymour Papert. It features "breaking programs into mind sized bites" (Papert, 1980).

Middle School--Grades 6-8.

"n"--Statistically, the number of cases in an array of cases.

One-Tailed Test of the Null Hypothesis--When a research hypothesis predicts the direction of significant difference (Isaac & Michael, 1985).

Pascal--A high level computer programming language named after the 17th century mathematician, Blaise Pascal. The language features modularization, i.e., reducing programs to easily solvable sub-units.

t-test--A statistical procedure to determine whether a significant difference exists between two means.

Two-Tailed Test of the Null Hypothesis--When a research hypothesis predicts that a difference will exist in the data and allows this difference to be in either direction (Isaac & Michael, 1985).

Watson-Glaser, Critical Thinking Appraisal--A standardized/normed test designed to measure important abilities involved in critical thinking. The assessment consists of five subtests, inference, recognition of assumptions, deduction, interpretation, and evaluation of arguments.

VI. LIMITATIONS OF THE STUDY

1. Conclusions resulting from analysis of the data generated by this study must be viewed as tenuous and exploratory in nature, because of less than desirable cell sizes. While authorities often differ regarding minimum acceptable "n" (Robert J. Duby, Personal Communication, May 1, 1988; Borg & Gall, 1979; Gage, 1963), clearly the majority agrees that large samples provide more strength of prediction and accuracy of inferred results than small samples (Best, 1977; Rummell, 1964). As Richard Runyon and Audrey Haber (1967: 192) state, "the power of a test is a function of N." As proposed, the project was to have a minimum n of 50 per cell. Unfortunately, much data was lost in transport (76 cases). Hence, cell sizes have been reduced to twenty-one for the Pascal Group, fifty-two for the BASIC Group, and forty for Non-programming Group. However, as Robert Winkler and William Hays (1971) explain, many statistical studies are exploratory in nature and do not require extensive sample size to investigate associations and map out the main relationships in a specified area.

2. The study applies to high school students only. No attempt should be made to generalize the results to any other group.
3. The study applies to the programming languages BASIC and Pascal. No attempt should be made to generalize the results to students programming in other languages without cautious analysis and further testing.
4. Less than desirable testing conditions may have existed because the author was not present at the test sites.
5. The study was conducted using human subjects; therefore it is impossible to insure that all groups were completely equivalent.

VII. BASIC ASSUMPTIONS

1. Students in the programming classes were above average in academic ability and could be compared to above-average students in non-programming classes.
2. Participating students had equivalent subject matter backgrounds.
3. All participating students in programming classes had equal access to computers.
4. All participating students had equal instruction time.
5. For this study, the Watson-Glaser Critical Thinking Appraisal was a valid and reliable measure of critical thinking skills.
6. Development of effective critical thinking skills results in increased problem solving ability.

VIII. DATA COLLECTION PROCEDURES

The target population was identified with the cooperation of Nova University Doctoral Students in Computer Education, and members of the Florida Instructional Computing Supervisors. Criteria for participation was enrollment in a high school that (1) taught both BASIC and Pascal, (2) allowed students to take Pascal without first having BASIC, and (3) had the same entry requirements for both Pascal and BASIC. Beginning in March, 1988, a set of Watson-Glaser Critical Thinking Appraisal Tests, complete with directions and answer sheets, was mailed to all participating schools. Teachers were asked to administer the 40-minute test to:

1. students taking a course in Pascal as their first programming experience.
2. a similar number of students taking BASIC as their first programming experience.
3. a similar number of above-average students having experienced no programming in high school. All students tested in this group had scores in the 55th to 85th percentile range on

the standardized test instrument used at the school. Selection in this manner helped ensure group equivalency because, as Henry Becker (1987:160) found in a nationwide survey,

> Most of the programming classes in high school are composed of students typically above average in ability but only a very small minority are composed primarily of very high ability students.

IX. DATA TREATMENT PROCEDURES

After collecting all data, the means of the BASIC Programming Group, the Pascal Programming Group, the combined Pascal and BASIC Programming Group (hereafter referred to as the Programming Group), and the Non-programming Group were calculated. Said means were ciphered for the total battery and each of the five subtests.

Using a one-tailed t-test, the six means of the Programming Group were compared to the corresponding means of the Non-programming Group to determine whether the Programmers scored significantly higher than their Non-programming counterparts. On this base, the null hypothesis (Ho), students participating in programming classes do not exhibit better critical thinking skills than comparable students not participating in programming classes, was tested for rejection or acceptance. Following an identical procedure, the means of the BASIC Programmers were compared to the Non-programmers, and the means of the Pascal Group were compared to the Non-programming Group. Based on the t-ratios, the null hypotheses that state (1) students participating in BASIC programming classes do not exhibit better

critical thinking skills than comparable students not participating in a BASIC class, and (2) students paticipating in a Pascal programming class do not exhibit better critical thinking skills than comparable students not participating in a Pascal class, were tested for rejection or acceptance.

Next, the results of a two-tailed t-test were used to compare the six means of the Pascal and BASIC Groups. Said results were used to determine whether a significant difference exists in critical thinking skill development due to participation in BASIC or Pascal programming classes. As a result, the null hypothesis, no significant difference exists in critical thinking skill development between students participating in BASIC or Pascal classes was tested for rejection or acceptance.

X. PRESENTATION OF RESULTS

The data for each participant include five subtests and a total battery score on the Watson-Glaser Critical Thinking Appraisal. Fifty-two BASIC programming students, twenty-one Pascal programming students, and forty Non-programming students participated. Tables 1, 2, and 3 present the scores for each group. At the base of each table, the means, standard deviations, and n's for the subtests and total battery are displayed.

Table 4 presents descriptive statistics (means, standard deviations, and n's) for the Programming Group. This cluster was created by combining the scores of the BASIC Group (n=52) and the Pascal Group (n=21).

Table 5 exhibits the t-ratios generated from all data. Subtests one through five and the total battery are presented comparing (1) Programming and Non-programming Groups, (2) BASIC and Non-programming Groups, (3) Pascal and Non-programming Groups, and (4) BASIC and Pascal Programming Groups.

Table 1

A Summary of Scores for the BASIC Programming Students on the Watson-Glaser Critical Thinking Appraisal

Stu#	Sub Test1	Sub Test2	Sub Test3	Sub Test4	Sub Test5	Total Battery
1	8	12	9	8	12	49
2	10	10	9	11	11	51
3	9	10	9	9	9	46
4	9	6	12	9	13	49
5	8	9	8	11	10	46
6	8	12	11	11	10	52
7	10	8	10	13	13	54
8	3	6	11	12	8	40
9	8	10	6	10	12	46
10	13	14	16	14	13	70
11	8	7	11	15	14	55
12	3	8	10	7	7	35
13	10	6	8	12	9	45
14	13	15	10	12	11	61
15	14	14	12	15	14	69
16	10	8	9	13	12	52
17	8	13	12	13	13	59
18	4	9	9	13	14	49
19	10	14	11	11	14	60
20	7	12	7	12	13	51
21	8	5	9	12	10	44
22	8	11	11	12	13	55
23	11	6	11	12	14	54
24	6	12	13	13	12	56
25	6	12	6	8	13	45
26	10	13	11	12	13	59
27	9	9	14	13	13	58
28	7	7	9	12	13	48
29	10	11	10	8	11	50
30	13	9	10	13	14	59
31	9	14	14	13	13	63
32	6	13	8	9	15	51
33	7	15	15	12	11	60
34	8	11	12	14	13	58
35	8	12	11	16	13	60
36	7	13	10	14	13	57

Table 1 (continued)

A Summary of Scores for the BASIC Programming Students on the Watson-Glaser Critical Thinking Appraisal

Stu#	Sub Test1	Sub Test2	Sub Test3	Sub Test4	Sub Test5	Total Battery
37	7	13	15	12	12	59
38	11	14	13	13	15	66
39	12	14	13	14	12	65
40	14	11	11	13	15	64
41	9	11	9	13	14	56
42	8	12	6	14	13	53
43	7	8	7	13	11	46
44	8	11	10	10	8	47
45	11	15	14	16	8	64
46	4	9	7	10	9	39
47	9	7	7	16	13	52
48	10	13	7	10	13	53
49	10	5	12	16	15	58
50	7	5	5	12	11	40
51	12	10	12	10	0	44
52	10	12	10	10	10	52
M	8.750	10.500	10.231	12.038	12.059	53.346
SD	2.520	2.927	2.541	2.178	2.014	7.854
n	52	52	52	52	51	52

Table 2

A Summary of Scores for the Pascal Programming Students on the Watson-Glaser Critical Thinking Appraisal

Stu#	Sub Test1	Sub Test2	Sub Test3	Sub Test4	Sub Test5	Total Battery
1	2	7	9	7	10	35
2	8	12	10	10	13	53
3	8	10	10	9	12	49
4	9	6	7	11	10	43
5	6	8	9	13	13	49
6	3	8	11	11	9	42
7	9	13	12	11	9	54
8	9	13	10	14	13	59
9	9	11	13	15	13	61
10	8	14	8	13	13	56
11	10	7	12	12	7	48
12	11	16	16	15	14	72
13	7	12	7	9	7	42
14	8	11	9	9	9	46
15	4	9	9	10	12	44
16	10	7	10	11	13	51
17	12	8	9	14	12	55
18	9	13	10	10	9	51
19	8	14	14	15	14	65
20	5	10	12	13	14	54
21	6	10	15	15	12	58
M	7.667	10.429	10.571	11.762	11.333	51.762
SD	2.556	2.803	2.441	2.385	2.266	8.584
n	21	21	21	21	21	21

Table 3

A Summary of Scores for the Non-Programming Students on the Watson-Glaser Critical Thinking Appraisal

Stu#	Sub Test1	Sub Test2	Sub Test3	Sub Test4	Sub Test5	Total Battery
1	7	13	12	10	15	57
2	6	9	8	10	6	39
3	5	11	6	14	6	42
4	6	13	6	16	10	51
5	6	4	7	11	10	38
6	5	8	8	11	9	41
7	7	11	10	15	9	52
8	6	10	8	10	12	46
9	7	9	11	6	10	43
10	10	12	11	11	15	59
11	12	15	13	10	12	62
12	8	11	7	13	10	49
13	3	6	10	10	10	39
14	4	6	10	14	12	46
15	10	13	9	10	11	53
16	3	13	12	8	7	43
17	5	13	7	8	13	46
18	4	12	4	9	7	37
19	8	8	11	12	12	51
20	6	12	7	12	6	43
21	4	11	8	12	5	40
22	5	13	9	10	11	48
23	8	13	11	15	14	61
24	6	5	10	13	13	45
25	6	7	10	13	12	48
26	6	15	4	12	11	48
27	10	11	11	15	14	61
28	7	12	11	11	11	52
29	13	8	9	11	12	53
30	11	10	9	8	11	49
31	3	8	8	10	10	39
32	9	8	11	13	13	54
33	7	5	9	12	11	44
34	7	7	10	6	11	41
35	6	9	8	9	9	41
36	12	4	11	7	7	41
37	11	13	11	16	16	67

Table 3 (continued)

A Summary of Scores for the Non-Programming Students on the Watson-Glaser Critical Thinking Appraisal

Stu#	Sub Test1	Sub Test2	Sub Test3	Sub Test4	Sub Test5	Total Battery
38	10	5	10	13	13	51
39	7	7	9	7	11	41
40	9	11	13	12	14	59
M	7.125	9.775	9.225	11.125	10.775	48.000
SD	2.604	3.084	2.154	2.631	2.684	7.656
n	40	40	40	40	40	40

Table 4

A Summary of Descriptive Statistics for the Programming Students on the Watson-Glaser Critical Thinking Appraisal

Stat	Sub Test1	Sub Test2	Sub Test3	Sub Test4	Sub Test5	Total Battery
M	8.438	10.479	10.329	11.959	11.685	52.890
SD	2.560	2.873	2.500	2.266	2.505	8.043
n	73	73	73	73	73	73

Table 5

A Summary of t-ratios Comparing the Means of the Programming, BASIC, Pascal, and Non-Programming Groups

	Group1*	Group2**	Group3***	Group4****
Subtest1	2.592	3.023	0.777	1.656
Subtest2	1.214	1.151	0.811	0.096
Subtest3	2.353	2.009	2.215	0.524
Subtest4	1.789	1.829	0.931	0.478
Subtest5	1.800	2.607	0.813	1.339
Tot Batt.	3.073	3.272	1.749	0.760

*Group1 = Programmers (despite language) compared to Non-programmers.
**Group2 = BASIC Programmers compared to Non-programmers.
***Group3 = Pascal Programmers compared to Non-programmers.
****Group4 = BASIC Programmers compared to Pascal Programmers.

XI. CONCLUSIONS AND SIGNIFICANCE

Before proceeding, it is necessary to focus again the reader's attention to the caveat presented on page thirteen. Because of small cell sizes, particularly in the Pascal Programming Group, the results of this study should be regarded as suspect and conclusions used as a guidance tool primarily to suggest interesting areas of further study.

For all t-test computations, the .05 level of confidence was used and the degrees of freedom (df) (n1 + n2 - 2) exceeded thirty. Therefore the values used to determine significance were:

1. >= 1.65 or <= -1.65 for Groups one, two, and three (one-tailed tests).
2. >= 1.96 or <= -1.96 for Group four (a two-tailed test).

Based on the t-ratios derived from total battery data, four conclusions are offered.

1. The Programming Group performed significantly better than the Non-programming Group on the Watson-Glaser. 3.073 > 1.65. Therefore, the null hypothesis (Ho) that states, students participating in programming classes do not

exhibit better critical thinking skills than comparable students not participating in programming classes, was rejected.

2. The BASIC Programming Group performed significantly better than the Non-programming Group. 3.272 > 1.65. Therefore, the Ho, students participating in BASIC programming classes do not exhibit better critical thinking skills than comparable students not participating in a BASIC class, was rejected.
3. The Pascal Programming Group performed significantly better than the Non-programming Group. 1.749 > 1.65. Therefore the Ho, students participating in Pascal programming classes do not exhibit better critical thinking skills than comparable students not particlpaing in Pascal classes, was rejected.
4. There was no significant difference in the scores of the BASIC Programming Group, and the Pascal Programming Group on the Watson-Glaser (total battery). 0.760 < 1.96. The Ho, no significant difference exists between students participating in BASIC and Pascal classes in problem solving skill development, was not rejected.

The following conclusions are based on the t-ratios generated from subtest data.

1. The Programming Group and the BASIC Group scored significantly higher (2.592 and 3.023 > 1.65) than the Non-programming Group on the Inference Test (Subtest 1). However, the Pascal Group did not score significantly higher (0.777 < 1.65).
2. There was no significant difference between the scores of any programming groups and the Non-programming Group on the Test of Assumption Recognition (subtest 2) (1.214, 1.151, and 0.811 < 1.65).
3. All programming groups scored significantly better than the non-programming Group on the Test of Deduction (subtest 3) (2.353, 2.009, and 2.215 > 1.65).
4. On the Test of Interpretation (subtest 4), the Programming Group and the BASIC Group scored significantly better than the Non-programming Group (1.789 and 1.829 > 1.165). However, the Pascal group did not (0.931 < 1.65).
5. The same pattern was repeated for the Evaluation of Arguments Test. The Programming and BASIC Groups scored significantly better

(1.800, 2.607 > 1.65), while the Pascal Group did not (0.813 < 1.65).

6. There was no significant difference between the BASIC and Pascal Programming Groups on any subtest of the Watson-Glaser (1.656, 0.096, 0.524, 0.478, and 1.339 < 1.96).

This study does not purport to provide definitive answers to many of the questions raised. However, the results do provide some guidance to educators (particularly computer educators) on several issues of concern.

First, it is clear that participation in programming classes may improve problem-solving skill development. On the total battery, all programming groups scored significantly higher than their non-programming counterpart.

Second, it appears that participation in programming classes may not affect all areas of problem solving equally. For example, no programming groups scored better than the Non-programming Group on the Recognition of Assumptions Test.

Third, it appears that the preference of computer educators for Pascal as the language to best develop problem-solving skills, instead of BASIC, may not be justified. On four of five subtests, students in the BASIC Programming Group scored significantly higher

than their non-programming counterparts. This event occurred only once with the Pascal Programming Group.

In summary, the most significant contributions of the project results are to:

1. lend support to the assumption that experiences in a computer programming class lead to increased problem-solving ability.
2. highlight some possibly inaccurate assumptions related to the choice of a computer language to best develop problem-solving skills in high school students.
3. suggest interesting and exciting areas for further exploration.

XII. FURTHER STUDIES

As a priority, this study must be replicated using larger correlated samples. Several iterations are needed. Many questions have been raised but final answers await more ambitious projects.

At the outset, four research hypotheses were proffered. Results indicate that one may be correct, i.e., there is no significant difference between the scores of students participating in a BASIC class and participants in a Pascal class on the Watson-Glaser Critical Thinking Appraisal. The use of alternative tests, measuring additional problem-solving skills, provides abundant opportunities for further study. Investigating the effect of other languages produces similar opportunities. Researchers may find that enrollment in other programming courses, such as C, or Logo, or Prolog, or some yet to be developed language, produces students with superior problem-solving skills.

Also, there is the question of the subtests. Each area investigated on the Watson-Glaser, plus others, provides intriguing subjects for additional research. Questions such as "why do BASIC programming students score significantly better than non-programming students on tests of inference but not assumption?" provide a myriad of possibilities.

Much needed investigation remains, requiring many projects to complete the task. This study and its predecessors will furnish the impetus for additional research to yield answers regarding the effect of computer programming on the development of generalized problem-solving skills in high school students.

REFERENCE LIST

Becker, Henry Jay. "Using Computers for Instruction." BYTE, 12(2):149-162. February, 1987.

Best, John W. RESEARCH IN EDUCATION, 3rd ed. Englewood Cliffs, N. J.: Prentice-Hall, Inc., 1977.

Beyer, Barry K. "Improving Thinking Skills - Practical Approaches." PHI DELTA KAPPAN, 65(8):556-560. April, 1984.

Beyer, Barry K. and Ronald E. Charlton. "Teaching Thinking Skills in Biology." AMERICAN BIOLOGY TEACHER, 48(4):207-212. April, 1986.

Borg, Walter R., and Meredith Damien Gall. EDUCATIONAL RESEARCH AN INTRODUCTION, 3rd ed. New York: Longman Publishers, 1979.

Burnett, Tom and Chuck Friesen. "Programming vs. Applications: Computer Curricula for K-12." T.H.E. JOURNAL. 14(3):84-87. October 1986.

Clements, Douglas H. "Programming, Problem Solving and Practice." ARITHMETIC TEACHER, 31(4)32-35. December, 1983.

Clements, Douglas H. "Longitudinal Study of the Effects of Logo Programming on Cognitive Abilities and Achievement." JOURNAL OF EDUCATIONAL COMPUTING RESEARCH. 3(1):73-94. January, 1987.

Clements, Douglas H. and Dominic F. Gullo. "Effects of Computer Programming on Young Children's Cognition." JOURNAL OF EDUCATIONAL PSYCHOLOGY, 76(6):1051-1058. December, 1984.

Dalbey, John and Marcia C. Linn. "The Demands and Requirements of Computer Programming: A Literature Review." JOURNAL OF EDUCATIONAL COMPUTING RESEARCH, 1(3):253-274. 1985.

Eckerman, David A. "Developing A Laboratory Course Using Pascal on the Apple II." In Thomas H. Beebe and Al P. Mizell (Eds), COMPUTER EDUCATION; BOOK OF READINGS (pp 209-213). Lexington, Massachusetts: Ginn Custom Publishing. September 1983.

Flavell, J. H. THE DEVELOPMENTAL PSYCHOLOGY OF JEAN PIAGET. Princeton, N. J.: D. VanNostrand Co., Inc., 1963.

Gage, N. L., ed. HANDBOOK OF RESEARCH ON TEACHING: A PROJECT OF THE AMERICAN EDUCATIONAL RESEARCH ASSOCIATION. Chicago: Rand McNally & Company, 1963.

Ginsberg, Herbert and Sylvia Opper. PIAGET'S THEORY OF INTELLECTUAL DEVELOPMENT. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1969.

Harvey, Brian. "Why Logo." In Thomas H. Beebe and Al P. Mizell (Eds), COMPUTER EDUCATION; BOOK OF READINGS (pp 163-172). Lexington, Massachusetts: Ginn Custom Publishing. September, 1983.

Hilgard, Ernest R., Richard C. Atkinson and Rita L. Atkinson. INTRODUCTION TO PSYCHOLOGY. New York: Harcourt, Brace & Jovanovich, 1975.

Hudgins, Bryce B. LEARNING AND THINKING: A PRIMER FOR TEACHERS. Itasca, Illinois: F. E. Peacock, 1977.

Hughes, Elizabeth. "A Beginner's Guide to BASIC Programming." In Thomas H. Beebe and Al P. Mizell (Eds), COMPUTER EDUCATION; BOOK OF READINGS (pp 189-196). Lexington, Massachusetts: Ginn Custom Publishing. September, 1983.

Issac, Stephen and William B. Michael. HANDBOOK IN RESEARCH AND EVALUATION. San Diego, California: EdITS Publishers, 1985.

Linn, Marcia C. "The Cognitive Consequences of Programming Instruction in Classrooms." EDUCATIONAL RESEARCHER, 14(5):14-16 and 25-29, May, 1985.

Luehrmann, Arthur. "Why Teach Programming: Two Bad Arguments." ELECTRONIC LEARNING, 4(7):14. April, 1985.

Markuson, Carolyn, Joyce Tobias, and Tom Lough. "Logo Fever: The Computer Language Every School Is Catching." INSTRUCTOR XCII(5):74-76 & 79, January, 1983.

Papert, Seymour. MINDSTORMS. New York: Basic Books, Inc., 1980.

Papert, Seymour. "Computer Criticism and Technocentric Thinking." EDUCATIONAL RESEARCHER, 16(1):22-30. January-February, 1987.

Perry, Margaret. "Developing Thinking Skills in Middle School Students . . . Using Computers." NAASP Bulletin, 70(493):97-100. November, 1986.

Pintrich, Paul R., Carl F. Berger, and Paul Stemmer. "Student's Programming Behavior in a Pascal Course." JOURNAL OF RESEARCH IN SCIENCE TEACHING, 24(5):451-477. May, 1987.

Presley, Bruce, and Tim Corica. A GUIDE TO PROGRAMMING IN TURBO PASCAL. Albany, New York: Delmar Publishers, Inc., 1986.

Rummel, J. Francis. AN INTRODUCTION TO RESEARCH PROCEDURES IN EDUCATION, 2nd ed. New York: Harper & Row, Publishers, 1964.

Runyon, Richard P., and Audrey Haber. FUNDAMENTALS OF BEHAVIORAL STATISTICS. Reading, MA: Addison-Wesley Publishing Company, 1967.

Saylor, J. Galen, and William M. Alexander. CURRICULUM PLANNING FOR MODERN SCHOOLS. New York: Holt, Rinehart and Winston, Inc., 1966.

Schank, Roger C. THE COGNITIVE COMPUTER. Reading, Massachusetts: Addison-Wesley Publishing Company, Inc., 1984.

Shelly, Gary B., Thomas J. Cashman, and Steven G. Forsythe. TURBO PASCAL PROGRAMMING. Boston: Boyd and Fraser Publishing Company, 1987.

Solomon, Cynthia. "Introducing Logo to Children." In Thomas H. Beebe and Al P. Mizell (Eds), COMPUTER EDUCATION; BOOK OF READINGS (pp 173-187). Lexington, Massachusetts: Ginn Custom Publishing. September, 1983.

Staff. "Boston Backs Basic." CLASSROOM COMPUTER LEARNING, 8(3):8-9. November/December, 1987.

Sternberg, Robert J. "Teaching Critical Thinking, Part 1: Are We Making Critical Mistakes?" PHI DELTA KAPPAN, 63(3):194-198. November, 1985.

Stronge, James. "Think of This." ELECTRONIC EDUCATION, 6(5):18-19. February, 1987.

Winkler, Robert L., and William L. Hays. STATISTICS: PROBABILITY, INFERENCE, AND DECISION, 2nd ed. New York: Holt, Rinehart and Winston, 1971.

APPENDIX A

COMPUTER LANGUAGE SURVEY

Please respond to the following questions by placing a check next to the response(s) that in your opinion apply.

1. Do any high schools in your district offer both BASIC and Pascal?
   ___ yes
   ___ no

2. If you answered yes to question #1, are students allowed to enroll in Pascal without having had BASIC?
   ___ yes
   ___ no
   ___ I do not know

3. In your opinion, what is the most compelling reason(s) for teaching programming in high school?
   ___ To develop job skills.
   ___ To develop problem solving/critical thinking skills.
   ___ To enable students to earn a high school credit.
   ___ To prepare students to excel in college programming classes.
   ___ To prepare students to excel in math, science and engineering programs in college.
   ___ To enable students to pass the AP test in computer science.
   ___ All the above.
   ___ None of the above.

   If your response to this question is "none of the above," what in your opinion is the most compelling reason for teaching programming in high school?

   _____

4. If a high school student could take only one programming course, which would you recommend to increase ability in problem solving and critical thinking.

___ BASIC
___ Pascal

5. If a high school student could take two programming courses, which would you recommend?

___ Introduction to BASIC then Advanced BASIC.
___ Introduction to BASIC then Introduction to Pascal.
___ Introduction to Pascal then Advanced Pascal.
___ Introduction to Pascal then Introduction to BASIC.